



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Generalized Filtering Decomposition*

Laura Grigori — Frederic Nataf

N° 7569

Janvier 2013

Thème NUM

*R*apport  
de recherche

arXiv:1103.3026v1 [cs.NA] 15 Mar 2011



# Generalized Filtering Decomposition

Laura Grigori\* , Frederic Nataf †

Thème NUM — Systèmes numériques  
Équipe-Projet Grand-large

Rapport de recherche n° 7569 — Janvier 2013 — 8 pages

**Abstract:** This paper introduces a new preconditioning technique that is suitable for matrices arising from the discretization of a system of PDEs on unstructured grids. The preconditioner satisfies a so-called filtering property, which ensures that the input matrix is identical with the preconditioner on a given filtering vector. This vector is chosen to alleviate the effect of low frequency modes on convergence and so decrease or eliminate the plateau which is often observed in the convergence of iterative methods. In particular, the paper presents a general approach that allows to ensure that the filtering condition is satisfied in a matrix decomposition. The input matrix can have an arbitrary sparse structure. Hence, it can be reordered using nested dissection, to allow a parallel computation of the preconditioner and of the iterative process.

**Key-words:** linear solvers, Krylov subspace methods, preconditioning, filtering property, block incomplete decomposition

\* INRIA Saclay - Ile de France, Laboratoire de Recherche en Informatique Université Paris-Sud 11, France ([laura.grigori@inria.fr](mailto:laura.grigori@inria.fr)).

† Laboratoire J.L. Lions, CNRS UMR7598, Université Paris 6, France ([nataf@ann.jussieu.fr](mailto:nataf@ann.jussieu.fr)).

# Décomposition à base de filtrage généralisée

**Résumé :** Ce document présente une nouvelle technique de préconditionnement adapté pour les matrices issues de la discrétisation d'un système d'équations aux dérivées partielles sur des maillages non structurés. Le préconditionneur satisfait une propriété dite de filtrage, qui signifie que la matrice d'entrée est identique au préconditionneur pour un vecteur donné de filtrage. Le choix de ce vecteur permet d'atténuer l'effet des modes de basse fréquence sur la convergence et ainsi de diminuer ou d'éliminer le plateau qui est souvent observé dans la convergence des méthodes itératives. En particulier, le document présente une approche générale qui permet d'assurer que la propriété de filtrage est satisfaite lors d'une décomposition matricielle. La matrice d'entrée peut avoir une structure creuse arbitraire. Ainsi, elle peut être rénumérotée en utilisant la méthode de dissection emboîtée, afin de permettre un calcul parallèle du préconditionneur et du processus itératif.

**Mots-clés :** solveurs linéaires, méthode de sous-espaces de Krylov, préconditionnement, propriété de filtrage, décomposition incomplète par blocs

# 1 Introduction

Iterative methods are widely used in industrial applications, and preconditioning these methods is an important topic which has already been extensively studied [5, 9, 3]. In this context, algebraic multigrid methods are very successful for certain classes of applications, in particular scalar PDEs [8, 11, 14, 4, 15]. They are known to have good weak scalability properties, but they are not strongly scalable. This motivates research on iterative solvers for systems of PDEs and/or large number of processors.

Several highly used preconditioners, as the incomplete LU factorizations and domain decomposition methods, are known to have scalability problems, in terms of both problem size and number of processors. This is often due to the presence of several low frequency modes that hinder the convergence of the iterative method. To solve this problem, a different class of so called filtering preconditioners has been proposed [1, 2, 16, 17], where the choice of the filtering vector is made to alleviate the effect of low frequency modes on the convergence. For domain decomposition methods, coarse grid correction is known to be mandatory for solving the scalability problem [13].

In this paper we focus on the generalization and suitability for parallel computing of the filtering preconditioner. This preconditioner is an incomplete factorization where it is possible to ensure that the factorization will coincide with the original matrix for some specified vector, called a filtering vector. Satisfying this filtering condition is an important factor for accelerating the convergence of the iterative method. The previous research on these methods considered only matrices arising from the discretization of PDEs on structured grids, where the matrix has a block tridiagonal structure [1, 2, 16, 17]. To the best of our knowledge, there was no previous result on the parallelization of filtering preconditioners. One of the important results of this research is the development of a new and general approach to ensure that a filtering condition is satisfied in a matrix decomposition. This approach is based on an innovative way of organizing the computations that allows on one side to satisfy a filtering property and on another side to perform a parallel computation. This approach has been used to develop a preconditioner based on a block approached decomposition, that we refer to as block filtering preconditioner. While we discuss in detail the right filtering property  $At = Mt$ , a similar approach can be used to develop a preconditioner that satisfies the left filtering property  $t^T A = t^T M$ , where  $A$  is the input matrix,  $M$  is the preconditioner and  $t$  is the filtering vector.

This preconditioner does not impose any particular structure on the input matrix. To allow its usage on parallel architectures, the input matrix can be reordered using nested dissection. This reordering allows a parallel implementation of the construction of the preconditioner, as well as of the iterative process.

The preconditioner can be seen as a generalization for unstructured grids of the preconditioner presented in [2] for block tridiagonal matrices. In contrast to the preconditioner presented in [2] that has been shown to be efficient in combination with ILU0, the block preconditioner presented here is efficient as a stand-alone preconditioner.

The goal of this paper is only to present the algebraic framework which allows a filtering condition to be satisfied in an incomplete block factorization. The numerical results showing the efficiency of the proposed preconditioner and its parallel performance will be presented in a future paper.

# 2 Block Filtering Decomposition

In this section we describe a block filtering preconditioner  $M$  which satisfies the right filtering condition  $(M - A)t = 0$ , where  $t$  is a filtering vector.

Consider a matrix  $A$  of size  $n \times n$  partitioned into a block matrix of size  $N \times N$  with square diagonal blocks (not necessarily of a same size)

$$A = \begin{pmatrix} A_{11} & \dots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \dots & A_{NN} \end{pmatrix}.$$

An exact block  $LDU$  factorization of  $A$  is written as

$$A = \begin{pmatrix} D_{11} & & & \\ L_{21} & D_{22} & & \\ \vdots & \ddots & \ddots & \\ L_{N1} & \dots & L_{N,N-1} & D_{NN} \end{pmatrix} \begin{pmatrix} D_{11}^{-1} & & & \\ & D_{22}^{-1} & & \\ & & \ddots & \\ & & & D_{NN}^{-1} \end{pmatrix} \begin{pmatrix} D_{11} & U_{12} & \dots & U_{1N} \\ & \ddots & \ddots & \vdots \\ & & D_{N-1,N-1} & U_{N-1,N} \\ & & & D_{NN} \end{pmatrix}, \quad (1)$$

where  $D_{ii}, i = 1 \dots N$  are square invertible matrices of size  $b_i \times b_i$  with  $b_i < n$ . Let  $D = \text{Block-Diag}(D_{11}, \dots, D_{NN})$ , and let

$$L = \begin{pmatrix} 0 & & & \\ L_{11} & \ddots & & \\ \vdots & \ddots & \ddots & \\ L_{N1} & \dots & L_{N,N-1} & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & U_{11} & \dots & U_{1,N} \\ & \ddots & \ddots & \vdots \\ & & \ddots & U_{N-1,N} \\ & & & 0 \end{pmatrix}.$$

The factorization (1) can be written as  $A = (L + D)D^{-1}(U + D)$ . In the following we refer to the blocks of  $L, D, U$  as  $C$ , where  $L_{ij} = C_{ij}$  if  $i > j$ ,  $U_{ij} = C_{ij}$  if  $i < j$ , and  $D_{ij} = C_{ij}$  if  $i = j$ . In other words, the matrix  $C$  can be written as  $C = L + D + U$ . The blocks of  $L, D$ , and  $U$  are computed using the following formula, with  $i, j = 1 \dots N$ :

$$C_{ij} = \begin{cases} A_{ij} & i = 1 \text{ or } j = 1 \\ A_{ij} - \sum_{k=1, L_{ik} \neq 0, U_{kj} \neq 0}^{\min(i,j)-1} L_{ik} D_{kk}^{-1} U_{kj}, & i > 1 \text{ or } j > 1 \end{cases} \quad (2)$$

In practice, even if the matrix  $A$  is very sparse, the factors  $L, D, U$  can be much denser. In particular the term  $L_{ik} D_{kk}^{-1} U_{kj}$  can introduce an important amount of fill-in in the factors. In our work, the goal is to approximate the inverse of the diagonal blocks  $D_{kk}^{-1}, k = 1 \dots n$  by a sparse matrix such that  $L_{ik} D_{kk}^{-1} U_{kj}$  stays sparse. In the context of filtering decomposition, there are mainly two approximations used. Consider a diagonal block  $D_{kk}$ . The first approach consists of approximating  $D_{kk}^{-1}$  by a sparse matrix  $\bar{F} = \beta$ , chosen such that a filtering condition is satisfied. The second approach aims at identifying a better approximation of  $D_{kk}^{-1}$  starting from  $\beta$ . As described in section 3, this leads to an approximation of the form  $\bar{F} = 2\beta - \beta D_{kk} \beta$  [2], where  $\beta$  is a diagonal matrix. We will discuss both approaches, but we note that the first approach is more stable and leads to better results in practice.

In the following we explain the construction of the block filtering preconditioner  $M$ . We first give its definition, and then explain more in detail the reasoning that lead to its construction. In section 3 we discuss the construction of the approximation  $\bar{F}$  of the inverse of the block diagonal matrices.

**Definition 2.1** Let  $A$  be a matrix of size  $n \times m$ . For  $k = 1 \dots N$ , let  $L_k$  be a matrix of size  $n \times n_k$ ,  $D_k$  be an invertible matrix of size  $n_k \times n_k$ , and  $U_k$  be a matrix of size  $n_k \times m$ . Let  $M$  be a matrix defined by

$$M - A = \sum_{k=1}^N L_k D_k^{-1} U_k - \sum_{k=1}^N L_k F_k U_k.$$

A construction that enables filtering is a construction where  $F_k, k = 1 \dots N$  are matrices that satisfy the relation

$$F_k U_k t = D_k^{-1} U_k t \quad \text{for all } k = 1 : N \quad (3)$$

**Definition 2.2** Let  $t$  be a filtering vector of size  $n$  and let  $A$  be a matrix of size  $n \times n$  partitioned into a block matrix of size  $N \times N$ . A block filtering decomposition is defined as

$$M = \begin{pmatrix} \bar{D}_{11} & & & \\ \bar{L}_{21} & \bar{D}_{22} & & \\ \vdots & \ddots & \ddots & \\ \bar{L}_{N1} & \dots & \bar{L}_{N,N-1} & \bar{D}_{NN} \end{pmatrix} \begin{pmatrix} \bar{D}_{11}^{-1} & & & \\ & \bar{D}_{22}^{-1} & & \\ & & \ddots & \\ & & & \bar{D}_{NN}^{-1} \end{pmatrix} \begin{pmatrix} \bar{U}_{1N} & & & \\ & \ddots & \ddots & \vdots \\ & & \bar{D}_{N-1,N-1} & \bar{U}_{N-1,N} \\ & & & \bar{D}_{NN} \end{pmatrix}, \quad (4)$$

where  $\bar{D}_{ii}, i = 1 \dots N$  are square invertible matrices of size  $b_i \times b_i$  with  $b_i < n$ . In more compact form,  $M = \bar{L} \bar{D} \bar{U}$ , where  $M, \bar{L}, \bar{D}, \bar{U}$  are block matrices of size  $N \times N$ . Let  $\bar{C} = \bar{L} + \bar{D} + \bar{U}$  and let  $t = (t_1; t_2; \dots t_N)$ . The blocks are computed as

$$\bar{C}_{ij} = \begin{cases} A_{ij} & i = 1 \text{ or } j = 1 \\ A_{ij} - \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} \bar{F}_{kj} \bar{U}_{kj}, & i > 1 \text{ or } j > 1 \end{cases} \quad (5)$$

where  $\bar{F}_{kj}$  is a sparse approximation of  $\bar{D}_{kk}^{-1}$  that satisfies

$$\bar{F}_{kj} \bar{U}_{kj} t_j = \bar{D}_{kk}^{-1} \bar{U}_{kj} t_j \quad \text{for all } k = 1 : \min(i, j) - 1 \text{ with } \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0 \quad (6)$$

If  $\bar{U}_{kj} t_j$  is a vector of nonzero elements, a matrix  $\bar{F}_{kj}$  that satisfies the condition in equation (6) can be computed as

$$\bar{F}_{kj} = \text{Diag}((\bar{D}_{kk}^{-1} \bar{U}_{kj} t_j) ./ \bar{U}_{kj} t_j)$$

where  $\cdot/$  is the pointwise division (see for example equation (15) in [2]). However, for very sparse matrices,  $\bar{U}_{kj}$  can have rows of all zeros, and hence the result of  $\bar{U}_{kj}t_j$  can be a vector with zero elements. We present in section 3 a construction of  $\bar{F}_{kj}$  that solves this problem.

The main idea in the design of the preconditioner in Definition 2.2 is to ensure that each block satisfies an appropriate filtering condition  $M_{ij}t_j = A_{ij}t_j$ , such that the global filtering condition  $Mt = At$  is satisfied, where  $t = (t_1; t_2; \dots t_N)$  is the filtering vector. We note  $B = M - A$ , and so we want to ensure that for each block  $B_{ij}t_j = 0$ . This is different from the approach used for block tridiagonal systems [2], where  $B = M - A$  is a block diagonal matrix. The matrix  $B = M - A$  is formed by  $(B_{ij})_{1 \leq i, j \leq N}$ , with

$$B_{ij} = \bar{C}_{ij} + \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} \bar{D}_{kk}^{-1} \bar{U}_{kj} - A_{ij}. \quad (7)$$

The construction of  $M$  ensures that for each block  $B_{ij}$ , for each term  $\bar{L}_{ik} \bar{D}_{kk}^{-1} \bar{U}_{kj}$  of the summation in Equation 7,  $\bar{F}_{kj}$  is chosen such that the filtering is satisfied. That is,  $\bar{L}_{ik} \bar{D}_{kk}^{-1} \bar{U}_{kj}t_j = \bar{L}_{ik} \bar{F}_{kj} \bar{U}_{kj}t_j$ . From this the formula of  $\bar{F}_{kj}$  in Equation 6 is deduced. This ensures that the global filtering for the whole matrix is satisfied. Note that there is a  $\bar{F}_{kj}$  for each nonzero block  $\bar{U}_{kj}$ , that is the approximation of the diagonal block depends on the off-diagonal blocks of  $\bar{U}$ . We give a formal proof in the following lemma.

**Lemma 2.1** *Consider an  $n \times n$  matrix  $A$  and a filtering vector  $t$  of size  $n$ . If the block filtering preconditioner  $M$  as defined in Definition 2.2 exists, then it satisfies the filtering property, that is  $Mt = At$ .*

**Proof.** The preconditioner  $M$  satisfies the right filtering property if for each nonzero block  $B_{ij}$  we have  $B_{ij}t_j = 0$ , where  $B_{ij}$  is of size  $b_i \times b_j$  and  $t_j$  is a vector of  $b_j$  elements. In the formula of  $B_{ij}$  from equation (7), we replace the expression of  $\bar{C}_{ij}$  from equation (5). We obtain:

$$\begin{aligned} B_{ij}t_j &= \left( \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} \bar{D}_{kk}^{-1} \bar{U}_{kj} - \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} \bar{F}_{kj} \bar{U}_{kj} \right) t_j = \\ &= \left( \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} \bar{D}_{kk}^{-1} (I - \bar{D}_{kk} \bar{F}_{kj}) \bar{U}_{kj} \right) t_j = 0 \end{aligned}$$

■

We give now a definition of the block filtering preconditioner, in which the inverse of a diagonal block matrix  $\bar{D}_{kk}$  is approximated by  $2\bar{F}_{kj} - \bar{F}_{kj} \bar{D}_{kk} \bar{F}_{kj}$ . We show that if the matrix  $\bar{F}_{kj}$  satisfies the same condition as in equation (6), the preconditioner satisfies the filtering property.

**Definition 2.3** *A block filtering preconditioner  $M$  of a matrix  $A$  of size  $n \times n$  is defined by an incomplete block factorization*

$$M = \begin{pmatrix} \bar{D}_{11} & & & \\ \bar{L}_{21} & \bar{D}_{22} & & \\ \vdots & \ddots & \ddots & \\ \bar{L}_{N1} & \dots & \bar{L}_{N,N-1} & \bar{D}_{NN} \end{pmatrix} \begin{pmatrix} \bar{D}_{11}^{-1} & & & \\ & \bar{D}_{22}^{-1} & & \\ & & \ddots & \\ & & & \bar{D}_{NN}^{-1} \end{pmatrix} \begin{pmatrix} \bar{D}_{11} & \bar{U}_{12} & \dots & \bar{U}_{1N} \\ & \ddots & \ddots & \vdots \\ & & \bar{D}_{N-1,N-1} & \bar{U}_{N-1,N} \\ & & & \bar{D}_{NN} \end{pmatrix}. \quad (8)$$

and a filtering vector  $t$  of size  $n$ , where  $\bar{D}_{ii}, i = 1 \dots N$  are square invertible matrices of size  $b_i \times b_i$  with  $b_i < n$ . In more compact form,  $M = \bar{L} \bar{D} \bar{U}$ , where  $M, \bar{L}, \bar{D}, \bar{U}$  are block matrices of size  $N \times N$ . Let  $\bar{C} = \bar{L} + \bar{D} + \bar{U}$  and let  $t = (t_1; t_2; \dots t_N)$ . The blocks are computed as

$$\bar{C}_{ij} = \begin{cases} A_{ij} & i = 1 \text{ or } j = 1 \\ A_{ij} - \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} (2\bar{F}_{kj} - \bar{F}_{kj} \bar{D}_{kk} \bar{F}_{kj}) \bar{U}_{kj}, & i > 1 \text{ or } j > 1 \end{cases} \quad (9)$$

where  $\bar{F}_{kj}$  is a sparse approximation of  $\bar{D}_{kk}$  that satisfies

$$\bar{F}_{kj} \bar{U}_{kj} t_j = \bar{D}_{kk}^{-1} \bar{U}_{kj} t_j \quad (10)$$

**Lemma 2.2** *Consider an  $n \times n$  matrix  $A$  and a filtering vector  $t$  of size  $n$ . If the block filtering preconditioner  $M$  as defined in Definition 2.3 exists, then it satisfies the filtering property, that is  $Mt = At$ .*

**Proof.** We use the same approach as in Lemma 2.1. The preconditioner  $M$  satisfies the right filtering property if for each nonzero block  $B_{ij}$  we have  $B_{ij}t_j = 0$ . In the formula of  $B_{ij}$  from equation (7), we replace the expression of  $\bar{C}_{ij}$  from equation (9). We obtain:

$$\begin{aligned} B_{ij}t_j &= \left( \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{ki} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} \bar{D}_{kk}^{-1} \bar{U}_{kj} - \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{kj} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} (2\bar{F}_{kj} - \bar{F}_{kj} \bar{D}_{kk} \bar{F}_{kj}) \bar{U}_{kj} \right) t_j = \\ &= \left( \sum_{k=1, \bar{L}_{ik} \neq 0, \bar{U}_{ki} \neq 0}^{\min(i,j)-1} \bar{L}_{ik} (\bar{F}_{kj} \bar{D}_{kk} - I) \bar{D}_{kk}^{-1} (\bar{D}_{kk} \bar{F}_{kj} - I) \bar{U}_{kj} \right) t_j = 0 \end{aligned}$$

■

### 3 Construction of the approximation

We describe the construction of the approximation matrices  $\bar{F}_{kj}$ . We denote the element in position  $(i, j)$  of a matrix  $A$  as  $A(i, j)$  and the element in position  $i$  of a vector  $v$  as  $v(i)$ .

The block filtering preconditioner defined in Definitions 2.2 and 2.3 requires the construction of matrices  $\bar{F}_{kj}$  that satisfy the equation (6), that is  $\bar{F}_{kj} \bar{U}_{kj} t_j = \bar{D}_{kk}^{-1} \bar{U}_{kj} t_j$ . We note in the following  $M_{kj} t_j = v_{kj}$  and  $\bar{D}_{kk}^{-1} \bar{U}_{kj} t_j = u_{kj}$ , where  $v_{kj}, u_{kj}$  are vectors of  $b_k$  elements. Hence we have  $\bar{F}_{kj} v_{kj} = u_{kj}$ . In the following, for the ease of understanding, we simplify the notation and discuss the relation  $\bar{F}v = u$ . The approach used previously for the construction of  $\bar{F}$  is to compute it as

$$\bar{F} = \text{Diag}(u./v)$$

where  $./$  is pointwise division.

For sparse matrices, the vector  $v$  can have zero elements. Possibly  $u$  can have only zero elements, but this case is simple to solve. We discuss the case of  $v$  having zero elements. If  $v$  has only zero elements, then  $u$  is also zero, and hence the relation  $\bar{F}v = u$  is satisfied. We discuss hence the case when there is at least a nonzero element in  $v$ . Let  $j$  be the index of a nonzero element, that is  $v(j) \neq 0$ . If  $v(i) = 0$ , then we take  $\bar{F}(i, j) = u(j)/v(j)$ . In other words, a simple construction of the matrix  $\bar{F}$  is as follows:

$$\bar{F}(i, j) = \begin{cases} u(i)/v(i) & \text{if } i = j \text{ and } v(i) \neq 0 \\ u(i)/v(j) & \text{if } v(i) = 0 \text{ and } j = \min_{k: v(k) \neq 0} |k - i| \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

An example of construction of  $\bar{F}$  is as follows:

$$\begin{pmatrix} 0 & u(1)/v(2) & & & & \\ & u(2)/v(2) & & & & \\ & u(3)/v(2) & 0 & & & \\ & & & 0 & u(4)/v(5) & \\ & & & & u(5)/v(5) & \\ & & & & u(6)/v(5) & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ v(2) \\ 0 \\ 0 \\ v(5) \\ 0 \end{pmatrix} = \begin{pmatrix} u(1) \\ u(2) \\ u(3) \\ u(4) \\ u(5) \\ u(6) \end{pmatrix}$$

The matrix  $\bar{F}$  can be easily constructed to be symmetric by letting  $\bar{F}(j, i) = \bar{F}(i, j)$ . But  $\bar{F}$  might not be SPD.

We can also use deflation techniques [7, 10, 12, 6] to construct  $\bar{F}_{kj}$  that satisfies the equation (6), that is  $\bar{F}_{kj} \bar{U}_{kj} t_j = \bar{D}_{kk}^{-1} \bar{U}_{kj} t_j$ . Equation (12) defines  $\bar{F}_{kj}$  for a symmetric matrix.

$$\bar{F}_{kj} = P + Q \quad (12)$$

$$P = I - QA \quad (13)$$

$$Q = ZE^{-1}Z^T \quad (14)$$

$$E = (Z^T \bar{D}_{kk} Z)^{-1} \quad (15)$$

$$Z = \bar{U}_{kj} t_j \quad (16)$$



### 3.1 Suitability for Parallelism

The block filtering preconditioner was defined in a general way. With a suitable ordering, a parallel preconditioner can be obtained. In our work, we focus on matrices partitioned using nested dissection. This partitioning leads to algorithms that can be implemented in parallel. We describe here briefly this ordering. Nested dissection considers the undirected graph  $G$  of a symmetric matrix  $A$ . It identifies a separator  $S$  that partitions the graph into two disconnected graphs  $G_1, G_2$ . The input matrix is permuted such that the vertices corresponding to the separator  $S$  are ordered after the vertices corresponding to the two disconnected graphs  $G_1, G_2$ . Then the same partitioning can be applied on the two disconnected graphs, with the recursion being stopped when the number of desired independent parts has been reached.

Consider a matrix  $A$  of size  $n \times n$  partitioned using nested dissection into a block matrix of size  $N \times N$ . The following example displays the result obtained after applying two steps of nested dissection that leads to a block matrix of size  $7 \times 7$ .

$$PAP^T = \begin{pmatrix} A_{11} & A_{13} & & & & & A_{17} \\ & A_{22} & A_{23} & & & & A_{27} \\ A_{31} & A_{32} & A_{33} & & & & A_{37} \\ & & & A_{44} & & A_{46} & A_{47} \\ & & & & A_{55} & A_{56} & A_{57} \\ & & & A_{64} & A_{65} & A_{66} & A_{67} \\ A_{71} & A_{72} & A_{73} & A_{74} & A_{75} & A_{76} & A_{77} \end{pmatrix},$$

The preconditioner  $M$  is defined as

$$M = \begin{pmatrix} \bar{D}_{11} & & & & & & \\ & \bar{D}_{22} & & & & & \\ \bar{L}_{31} & \bar{L}_{32} & \bar{D}_{33} & & & & \\ & & & \bar{D}_{44} & & & \\ & & & & \bar{D}_{55} & & \\ \bar{L}_{64} & \bar{L}_{65} & \bar{D}_{66} & & & & \\ \bar{L}_{71} & \bar{L}_{72} & \bar{L}_{73} & \bar{L}_{74} & \bar{L}_{75} & \bar{L}_{76} & \bar{D}_{77} \end{pmatrix} (\bar{D}_{ii}^{-1})_{i=1:7} \begin{pmatrix} \bar{D}_{11} & & & & & & \bar{U}_{17} \\ & \bar{D}_{22} & & & & & \bar{U}_{27} \\ & & \bar{D}_{33} & & & & \bar{U}_{37} \\ & & & \bar{D}_{44} & & & \bar{U}_{46} \\ & & & & \bar{D}_{55} & & \bar{U}_{56} \\ & & & & & \bar{D}_{66} & \bar{U}_{67} \\ & & & & & & \bar{U}_{77} \end{pmatrix} \quad (17)$$

where each block of the factors  $\bar{L}, \bar{D}, \bar{U}$  can be computed following Definitions 2.2 or 2.3. With this partition, both the preconditioner and the iterative process can be implemented in parallel.

## 4 Conclusions

In this report we have briefly presented a block filtering preconditioner  $M$  that is build from an input matrix  $A$  and a filtering vector  $t$  and satisfies the property  $Mt = At$ . With an appropriate ordering as nested dissection, this preconditioner is suitable for parallel implementations. A future paper will focus on numerical results on scalar of PDEs discretized on two-dimensional and three-dimensional structured and unstructured grids showing that this method is efficient in practice.

## References

- [1] Y. Achdou and F. Nataf. An iterated tangential filtering decomposition. *Numer. Linear Algebra Appl.*, 10(5-6):511–539, 2003. Preconditioning, 2001 (Tahoe City, CA).
- [2] Y. Achdou and F. Nataf. Low frequency tangential filtering decomposition. *Numer. Linear Algebra Appl.*, 14(2):129–147, 2007.
- [3] Michele Benzi and Miroslav Tuma. A comparative study of sparse approximate inverse preconditioners. *Appl. Num. Math.*, 30:305–340, 1999.
- [4] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55(4):379–393, 1995.
- [5] G. Meurant. *Computer solution of large linear systems*. North-Holland Publishing Co., Amsterdam, 1999.
- [6] R. Nabben and C. Vuik. A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners. *Numer. Linear Algebra Appl.*, 15(4):355–372, 2008.

- 
- [7] R. A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24(2):355–365, 1987.
  - [8] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 73–130. SIAM, Philadelphia, PA, 1987.
  - [9] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.
  - [10] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc’h. A deflated version of the conjugate gradient algorithm. *SIAM J. Sci. Comput.*, 21(5):1909–1926 (electronic), 2000. Iterative methods for solving systems of algebraic equations (Copper Mountain, CO, 1998).
  - [11] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128(1-2):281–309, 2001. Numerical analysis 2000, Vol. VII, Partial differential equations.
  - [12] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *J. Sci. Comput.*, 39(3):340–370, 2009.
  - [13] A. Toselli and O. Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.
  - [14] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. Stüben.
  - [15] P. Vaněk, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numer. Math.*, 88(3):559–579, 2001.
  - [16] C. Wagner. Tangential frequency filtering decompositions for unsymmetric matrices. *Numer. Math.*, 78(1):143–163, 1997.
  - [17] C. Wagner and G. Wittum. Adaptive filtering. *Numer. Math.*, 78(2):305–328, 1997.



---

Centre de recherche INRIA Saclay – Île-de-France  
Parc Orsay Université - ZAC des Vignes  
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex  
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier  
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq  
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex  
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex  
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex  
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399